IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Searching and Recording Media Streams

Inventor(s):
Nosakhare D. Omoigui

ATTORNEY'S DOCKET NO. MS1-362US

TECHNICAL FIELD

This invention relates to networked client/server systems and to searching and recording streaming media content in such systems.

BACKGROUND OF THE INVENTION

Multimedia streaming—the continuous delivery of synchronized media data like video, audio, text, and animation—is a critical link in the digital multimedia revolution. Today, streaming media is primarily about video and audio, but a richer, broader digital media era is emerging with a profound and growing impact on the Internet and digital broadcasting.

Synchronized media means multiple media objects that share a common timeline. Video and audio are examples of synchronized media—each is a separate data stream with its own data structure, but the two data streams are played back in synchronization with each other. Virtually any media type can have a timeline. For example, an image object can change like an animated .gif file, text can change and move, and animation and digital effects happen over time. This concept of synchronizing multiple media types is gaining greater meaning and currency with the emergence of more sophisticated media composition frameworks implied by MPEG-4, Dynamic HTML, and other media playback environments.

The term "streaming" is used to indicate that the data representing the various media types is provided over a network to a client computer on a real-time, as-needed basis, rather than being pre-delivered in its entirety before playback. Thus, the client computer renders streaming data as it is received from a network server, rather than waiting for an entire "file" to be delivered.

Streaming multimedia content enables a variety of informational content that was not previously available over the Internet or other computer networks. Live content is one significant example of such content. Using streaming multimedia, audio, video, or audio/visual coverage of noteworthy events can be broadcast over the Internet as the events unfold. Similarly, television and radio stations can transmit their live content over the Internet.

However, one current problem with streaming multimedia content is that users are typically limited to accessing the multimedia content via common "shuttle controls" on a multimedia player, such as a play button, fast forward button, pause button, etc. Given that large amounts of data can be stored as multimedia content (e.g., individual presentations lasting for hours), such controls make it difficult for a user to locate the portions of the multimedia content that are of most interest to him or her.

An additional problem with streaming multimedia content is that the user must typically be connected to the same network as the server (e.g., the Internet) in order to receive the streaming multimedia content. If this connection is not maintained then the streaming of the multimedia content stops. This "continuous connection" limitation can be troublesome for many individuals, such as those using portable computers in locations that may not always have access to the appropriate network, or individuals who do not want to tie up a telephone line for their network connection while playing back the multimedia content.

The invention described below addresses these disadvantages, providing for the searching and recording of streaming media content.

SUMMARY OF THE INVENTION

In a networked client/server system, media content is streamed from the server to the client. A user of the client can search the media content to identify temporal locations that satisfy certain search criteria, and/or store the media content locally at the client for subsequent playback.

According to one aspect of the invention, indexes are maintained for each of different media streams that can be streamed to the client either individually or together for a multimedia presentation. The indexes store a correspondence between content for a media stream and temporal locations of that media stream. In response to a user search request, search criteria is compared to the appropriate index(es) to identify whether the search criteria matches any data in the index(es).

According to another aspect of the invention, in response to a user search request the search criteria from the search request is compared directly to the media stream data rather than to an index. This comparison is made to identify whether the search criteria matches any of the media stream data.

According to another aspect of the invention, if data matching the search criteria is found (either in an index or the media stream data), then the media server "seeks" to a temporal location of the media stream identified by the matching data. The server then proceeds to stream the media content to the client beginning at that temporal location.

According to another aspect of the invention, a search request and corresponding search criteria are compared to multiple media streams (either directly or indirectly via associated indexes). Thus, a single search request can be used to search through all of the media streams of a multimedia presentation.

According to another aspect of the invention, the multiple media streams of a multimedia presentation are streamed from the server to the client and stored locally by the client. A markup document, referencing the multiple media streams also stored locally at the client, is generated and stored at the client. Thus, a user can play back the locally stored multimedia presentation at a later time when not coupled to a network and thus not able to receive streaming media from the server.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings. The same numbers are used throughout the figures to reference like components and/or features.

Fig. 1 shows a client/server network system and environment in accordance with the invention.

Fig. 2 shows a general example of a computer that can be used in accordance with the invention.

Fig. 3 illustrates an exemplary client-server relationship for streaming data.

Fig. 4 is a flowchart illustrating an exemplary process for searching media streams in accordance with one implementation of the invention.

Fig. 5 is a flowchart illustrating another exemplary process for searching media streams in accordance with another implementation of the invention.

Figs. 6 and 7 are block diagrams illustrating the local storage of a multimedia presentation in accordance with one implementation of the invention.

Fig. 8 is a flowchart illustrating an exemplary process for recording a multimedia presentation in accordance with one implementation of the invention.

DETAILED DESCRIPTION

General Network Structure

Fig. 1 shows a client/server network system and environment in accordance with the invention. Generally, the system includes one or more (*m*) network multimedia server computers 100, one or more (*z*) index server computers 102, and one or more (*n*) network client computers 104. The computers communicate with each other over a data communications network, which in Fig. 1 includes a public network 106 such as the Internet. The data communications network might also include local-area networks and/or private wide-area networks. Server computers 100 and client computers 104 communicate with one another via any of a wide variety of known protocols, such as the Hypertext Transfer Protocol (HTTP).

Multimedia servers 100 have access to streaming media content in the form of different media streams. These media streams can be individual media streams (e.g., audio, video, graphical, text, etc.), or alternatively composite media streams including multiple such individual streams. Some media streams might be stored as files 108 in a database or other file storage system, while other media streams 110 might be supplied to the server on a "live" basis from other data source components through dedicated communications channels or through the Internet itself.

The media streams received from servers 100 are rendered at the client computers 104 as a multimedia presentation, which can include media streams from one or more of the servers 100. These different media streams can include one or more of the same or different types of media streams. For example, a multimedia presentation may include two video streams, one audio stream, and one stream of graphical images.

A user interface (UI) at the client computer 104 allows users to control the playback of the multimedia presentation, such as selecting which of multiple presentations to play back, controlling pausing of the playback, etc. The UI at client 104 further allows a user to input search criteria for searching one or more of the individual media streams available from a server 100, and to save the media streams of a multimedia presentation for subsequent playback when not coupled to network 106.

Index servers 102 optionally maintain indexes for the streaming media data available from servers 100. These indexes provide a correspondence between elements or objects of the media data streams and temporal locations of the media data streams. These indexes can be used for searching the media data streams, as discussed in more detail below. Alternatively, the indexes may be maintained at the media servers 100.

Streaming Media

In this discussion, streaming media refers to one or more individual media streams being transferred over a network to a client computer on an as-needed basis rather than being pre-delivered in their entirety before playback. Each of the individual media streams corresponds to and represents a different media type and each of the media streams can be rendered by a network client to produce a user-perceivable presentation using a particular presentation medium. The individual media streams can be rendered to produce a plurality of different types of user-perceivable media, including synchronized audio or sound, video graphics or motion pictures, animation, textual content, command script sequences, or other media types that convey time-varying information or content in a way that can be sensed and perceived by a human. The individual media streams have their own timelines, which are synchronized with each other so that the media streams can be rendered simultaneously for a coordinated multimedia presentation. These individual media streams can be delivered to the client computer as individual streams from one or more servers, as a composite media stream(s) from one or more servers, or a combination thereof.

In this discussion, the term "composite media stream" describes synchronized streaming data that represents a segment of multimedia content. The composite media stream has a timeline that establishes the speed at which the content is rendered. The composite media stream can be rendered to produce a plurality of different types of user-perceivable media, such as synchronized audio or sound, video graphics or motion pictures, animation, textual content, command script sequences, etc. A composite media stream includes a plurality of individual media streams representing the multimedia content.

There are various standards for streaming media content and composite media streams. "Advanced Streaming Format" (ASF) is an example of such a standard, including both accepted versions of the standard and proposed standards for future adoption. ASF specifies the way in which multimedia content is stored, streamed, and presented by the tools, servers, and clients of various multimedia vendors. ASF provides benefits such as local and network playback, extensible media types, component download, scalable media types, prioritization of streams, multiple language support, environment independence, rich inter-stream relationships, and expandability. Further details about ASF are available from Microsoft Corporation of Redmond, Washington.

Regardless of the streaming format used, an individual data stream contains a sequence of digital data units that are rendered individually, in sequence, to produce an image, sound, or some other stimuli that is perceived by a human to be continuously varying. For example, an audio data stream comprises a sequence of sample values that are converted to a pitch and volume to produce continuously varying sound. A video data stream comprises a sequence of digitally-specified graphics frames that are rendered in sequence to produce a moving picture. An animation stream comprises a sequence of graphical images that are rendered in sequence to produce a moving image. An image stream comprises a sequence of graphical images that are rendered to produce a changing image over time. A text stream is a sequence of symbols and/or alphanumeric characters that are rendered to produce different symbol/character combinations over time (e.g., in the form of words).

For a composite media stream, the individual data streams are typically interleaved in a single sequence of data packets. Various types of data compression might be used within a particular data format to reduce communications bandwidth requirements.

The sequential data units (such as audio sample values, video frames, groups of characters, graphical images, etc.) of the individual streams are associated with both delivery times and presentation times, relative to an arbitrary start time. The delivery time of a data unit indicates when the data unit should be delivered to a rendering client. The presentation time indicates when the value should be actually rendered. Normally, the delivery time of a data unit precedes the presentation time.

The presentation times determine the actual speed of playback. For data streams representing actual events or performances, the presentation times correspond to the relative times at which the data samples were actually recorded. The presentation times of the various different individual data streams are consistent with each other so that the streams remain coordinated and synchronized during playback.

Exemplary Computer Environment

In the discussion below, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by one or more conventional personal computers. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. In a distributed computer environment, program modules may be located in both local and remote memory storage devices.

Alternatively, the invention could be implemented in hardware or a combination of hardware, software, and/or firmware. For example, one or more application specific integrated circuits (ASICs) could be programmed to carry out the invention.

Fig. 2 shows a general example of a computer 142 that can be used in accordance with the invention. Computer 142 is shown as an example of a computer that can perform the functions of any of server computers 100 or 102, or client computers 104 of Fig. 1.

Computer 142 includes one or more processors or processing units 144, a system memory 146, and a bus 148 that couples various system components including the system memory 146 to processors 144.

The bus 148 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) 150 and random access memory (RAM) 152. A basic input/output system (BIOS) 154, containing the basic routines that help to transfer information between elements within computer 142, such as during start-up, is stored in ROM 150. Computer 142 further includes a hard disk drive 156 for reading from and writing to a hard disk, not shown, a magnetic disk drive 158 for reading from and writing to a removable magnetic disk 160, and an optical disk drive 162 for reading from or writing to a removable optical disk 164 such as a CD ROM or other optical media. The hard disk drive 156, magnetic disk drive 158, and optical disk drive 162 are connected to the system bus 148 by an SCSI interface 166 or some other appropriate interface. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for computer 142. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 160 and a removable optical disk 164, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs) read only memories (ROM), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 160, optical disk 164, ROM 150, or RAM 152, including an operating system 170, one or more application programs 172, other program modules 174, and program data 176. A user may enter commands and information into computer 142 through input devices such as keyboard 178 and pointing device 180. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are connected to the processing unit 144 through an interface 182 that is coupled to the system bus. A monitor 184 or other type of display device is also connected to the system bus 148 via an interface, such as a video adapter 186. In addition to the monitor, personal computers typically include other peripheral output devices (not shown) such as speakers and printers.

Computer 142 operates in a networked environment using logical connections to one or more remote computers, such as a remote computer 188. The remote computer 188 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computer 142, although only a memory storage device 190 has been illustrated in Fig. 2. The logical connections depicted in Fig. 2 include a local area network (LAN) 192 and a wide area network (WAN) 194. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet. In the described embodiment of the invention, remote computer 188 executes an Internet Web browser program such as the "Internet Explorer" Web browser manufactured and distributed by Microsoft Corporation of Redmond, Washington.

When used in a LAN networking environment, computer 142 is connected to the local network 192 through a network interface or adapter 196. When used in a WAN networking environment, computer 142 typically includes a modem 198 or other means for establishing communications over the wide area network 194, such as the Internet. The modem 198, which may be internal or external, is connected to the system bus 148 via a serial port interface 168. In a networked environment, program modules depicted relative to the personal computer 142, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Generally, the data processors of computer 142 are programmed by means of instructions stored at different times in the various computer-readable storage media of the computer. Programs and operating systems are typically distributed, for example, on floppy disks or CD-ROMs. From there, they are installed or loaded into the secondary memory of a computer. At execution, they are loaded at least partially into the computer's primary electronic memory. The invention described herein includes these and other various types of computer-readable storage media when such media contain instructions or programs for implementing the steps described below in conjunction with a microprocessor or other data processor. The invention also includes the computer itself when programmed according to the methods and techniques described below. Furthermore, certain sub-components of the computer may be programmed to perform the functions and steps described below. The invention includes such sub-components when they are programmed as described. In addition, the invention described herein includes data structures, described below, as embodied on various types of memory media.

For purposes of illustration, programs and other executable program components such as the operating system are illustrated herein as discrete blocks, although it is recognized that such programs and components reside at various times in different storage components of the computer, and are executed by the data processor(s) of the computer.

Searching Data Streams

As shown in Fig. 1, a network system in accordance with the invention includes network server(s) 100 from which a plurality of composite media streams are available. In some cases, the composite media streams are actually stored by server(s) 100. In other cases, server(s) 100 obtains the composite media streams from other network sources or devices.

The system also includes network clients 104. Generally, the network clients are responsive to user input to select or request identified multimedia presentations. In response to a request for a multimedia presentation, server(s) 100 streams the requested media stream(s) to the network client in accordance with some known format such as ASF. The client renders the data streams to produce the multimedia presentation.

Fig. 3 illustrates an exemplary client-server relationship for streaming data. A network server 100 is illustrated streaming a composite media stream 202 to a client 104. Alternatively, multiple servers 100 could be streaming individual or composite media streams to client 104. Additional control information 203 is also communicated between server 100 and client 104 to manage the streaming of composite media stream 202 to client 104.

When a user requests a particular composite media stream, client 104 requests the underlying media streams from the appropriate server(s) 100. This request can be from a standalone control application that is stored and executed at client 104, or alternatively an application that is hosted at a server 100 and is transmitted to client 104 for execution. For example, the control application could be hosted in a HTTP web page (maintained by either server 100 or another server coupled to the network) in accordance with Hypertext Markup Language (HTML) or Extended Markup Language (XML). The control application (whether standalone or server-hosted) includes identifier(s) of the composite media stream and/or the individual media streams of the multimedia presentation, and coordinates when and how they are presented at client 104.

Each media stream has a timeline, and the timelines of the individual streams are synchronized with each other so that the streams can be rendered in combination to produce coordinated multimedia content at the network client 104. A streaming module 205 in server 100 manages the streaming of the composite media stream to client 104 based at least in part on the delivery times of the data units in the composite media stream.

The client computer has a demultiplexer component 204 that receives the composite media stream and separates out the individual media streams from the composite format in which the data is streamed (such as ASF). This results in a video stream 206, an audio stream 208, a text stream 210, an image stream 212, and an animation stream 214.

The individual media streams are received from demultiplexer 204 by respective decoders 222, 224, 226, 228, and 230 that perform in accordance with the particular data format being employed. For example, the decoders might perform data decompression. The decoded streams are then provided to and received by respective renderers 234, 236, 238, 240, and 242. The rendering components 234 – 242 render the streams as the streams continue to be streamed from the network server 100.

Server 100 stores a composite media stream 248 (e.g., in accordance with ASF) including multiple individual media streams 250 – 258 for a multimedia presentation. The individual media streams are of different types, which in the illustrated embodiment are audio stream 250, video stream 252, image stream 254, text stream 256, and animation stream 258. A streaming module 205 manages, on behalf of server 100, the communication between server 100 and client 104, including the streaming of the composite media stream 202 and communication of control information to and from client 104.

Server 100 also stores indexes 260 – 268, each corresponding to one of the individual media streams 250 – 258. Indexes 260 – 268 can be part of the same composite media stream 248 as the individual media streams 250 – 258, or alternatively may be stored separately. In the illustrated example, each individual media stream 250 – 258 has a corresponding index 260 – 268, illustrated as audio index 260, video index 262, image index 264, text index 266, and animation index 268. Alternatively, one or more of the indexes 260 – 268 may be combined together.

Alternatively, some or all of indexes 260 – 268 may be stored at a remote server, such as index server 102 of Fig. 1. Or, in other alternate embodiments, some or all of indexes 260 – 268 may be stored at client 104 or transferred to client 104 for searching.

Each of the indexes 260 – 268 maintains a correspondence between a particular term or element of the associated media stream and a temporal location of the associated media stream. This correspondence identifies, for each term or element, a temporal location(s) of the associated media stream at which that term or element occurs. This correspondence can be maintained by storing multiple entries in the index, each entry including a term or element of the associated media stream and a temporal location(s) at which that term or element occurs. In one implementation these terms or elements are characters, words, symbols, or groups thereof. Alternatively the exact nature of these terms or elements may be dependent on the nature of the associated media stream. For example, text index 266 may include words or phrases as terms or elements, while audio index 260 may include digital representations of audio waveforms as terms or elements.

Indexes 260 – 268 can be generated in any of a wide variety of manners, including both manual and automatic generation. Manual generation can be performed by an individual (e.g., the author of the multimedia presentation) manually identifying the different terms or elements to index for each of the individual media streams and the temporal location(s) of each of these terms or elements. Automatic generation of the indexes can be performed by server 100 or another device. The manner in which the automatic generation is carried out is dependent on the nature of the associated individual media stream.

Text streams can be indexed based on different elements, such as characters, symbols, words, or groups thereof (e.g., phrases or sentences). In the illustrated example, server 100 (or other device generating the index) generates the index by identifying each of the elements in the text stream and their corresponding presentation times. As each element can occur multiple times in a text stream, multiple presentation times may be identified for the index.

Image streams can be indexed in a similar manner as text streams. Any of a variety of conventional pattern recognition techniques can be used to identify particular objects in the image stream or different characteristics of those objects (e.g., color). A textual description of each of these objects is included as an element in the index, along with its corresponding presentation time.

Animation streams can be indexed in an analogous manner as image streams. However, each object in the animation may have a range of presentation times corresponding to, for example, the object moving or changing locations over time. Thus, multiple presentation times (the presentation times for this range) may be associated with the object. The earliest presentation time in the range may be used in the index as the temporal location for this object, or alternatively another temporal location within the range may be used (e.g., the entire range, or alternatively a mid-range or "average" temporal range).

Audio streams can be similarly indexed. Conventional audio analysis techniques may be used to identify words or groups of words in the audio stream, based on the digital representation of the analog waveform of the audio content. These digital representations of words or groups of words are stored in the index along with their temporal locations in the audio content. Alternatively, conventional speech to text techniques can be used to convert the audio stream to words which can then be included in the index analogous to text streams discussed above.

Video streams can also be indexed. Using image analysis techniques similar to those for analyzing image streams, different objects within the video stream can be identified and included in the index. Alternatively, general video frame characteristics can be indexed, such as predominant colors within the frame. Objects in a video stream may correspond to a range of presentation times analogous to those in animation streams, and the temporal locations of these can be stored in the index in a manner analogous to those of animation streams.

A user at client 104 can search through the media streams 250 – 258 by submitting a search request and search criteria (e.g., particular words, or other text or symbols) to server 102. A search engine 270 receives such search requests and accompanying search criteria at server 102. Search engine 270, upon receipt of the search request, compares the search criteria to the entries in each of indexes 260 – 268 to determine whether any of the entries match or are satisfied by the search criteria. Thus, a single search request from the user can initiate searching of multiple individual media streams. Alternatively, the user may identify in the search request specific media streams that are to be searched, with the other media streams to be left unsearched.

Additionally, search engine 270 can search multiple different individual or composite streams located at server 100, regardless of whether the streams correspond to the same or different media presentations. Alternatively, search engine 270 may forward a search request to other servers 100 to identify matches with media streams stored at such servers. Thus, a single search request from the user can initiate searching of media streams of multiple different media presentations regardless of where the streams are stored.

Alternatively, search engine 270 may not use indexes 260 – 268. Rather, upon receipt of a search request search engine 270 can access the various steams 250 – 258 directly to identify terms or elements to compare to the search criteria. These terms or elements can be identified in any of the manners discussed above with reference to generating the indexes.

Situations can arise where the search criteria match multiple entries in the index. For example, a particular object may occur multiple times in an image stream and thus have multiple temporal locations associated with it in the index. Search engine 270 can identify one of these temporal locations to use as the result of the search process. In one implementation, if the media stream is currently being played back, then the current presentation time of the multimedia presentation is identified (e.g., from client 104). Search engine 270 then selects the next temporal location associated with the index entry that is after the current presentation time. Alternatively, search engine 270 may select the temporal location that is closest to the current presentation time as the result, or alternatively may use some other process for identifying one of the presentation times. Alternatively, search engine 270 may make such determinations based on the delivery times for the data units of the media stream rather than the presentation times.

Search engine 270 can take a variety of different actions when a term or element in the index matches the search criteria. For example, the search engine 270 may inform streaming module 205 to "seek" to that temporal location and begin streaming the media stream to client 104 at that temporal location. By way of another example, the matching entry and associated temporal locations) may be returned to client 104 and displayed to the user.

Additionally, the search process may make use of global variables. For example, a global character can be used to represent one or more characters, symbols, or words during the searching process.

Fig. 4 is a flowchart illustrating an exemplary process for searching media streams in accordance with one implementation of the invention. The process of Fig. 4 may be performed in software, firmware, hardware, or any combination thereof. Fig. 4 is described with additional reference to components in Figs. 1 and 3.

An index for one of the media streams of a media presentation is initially generated, step 302. Indexes for any additional media streams are also generated, steps 304 and 302. This index generation can be carried out by multimedia server 100, index server 102, or alternatively some other device (not shown) and subsequently transferred to server 100 or 102.

Once the indexes are generated, server 100 eventually receives a search request with accompanying search criteria, step 306. Search engine 270 compares the index(es) of the media stream(s) corresponding to the search request to the search criteria, step 308, and attempts to identify a match, step 310. If no match is identified, then an indication that the search failed is sent to the client, step 312. Alternatively, steps 308 – 310 may be carried out at an index server 102 or client 104 rather than media server 102.

However, if a match is identified then search engine 270 identifies a temporal location corresponding to the match (selecting one of multiple temporal locations if necessary), step 314. Streaming module 205 then streams, to client 104, the data for the media presentation starting at a location based on the identified temporal location, step 316. Streaming module 205 may stream the data beginning at the identified temporal location, or alternatively may "rewind" or "back up" to a temporal point prior to the temporal location. This "rewinding" may be of a fixed amount (e.g., three seconds), or alternatively may be based on pauses or breaks. For example, conventional audio or video analysis programs may be used to identify pauses or breaks in the speech or action in the multimedia presentation, and module 205 may search back through the multimedia presentation beginning at the identified temporal location to identify such a pause or break and begin streaming at that location. Alternatively, rather than "seeking" to the match location in step 316, other indications of a match may be provided to client 104.

In alternative implementations, the index generation of steps 302 – 304 is performed in response to a user search request rather than as an initialization process as illustrated in Fig. 4. For example, when a user submits a request for a multimedia presentation that has not been indexed, the index generation may occur in response to the request (either generating all of the indexes or only those necessary until a match with the search criteria is reached).

Alternatively, rather than performing any sort of "rewind" or "backing up" process during the search request, such rewinding can be performed at the time the index is generated. Thus, rather than storing a specific presentation time that corresponds to the time that a particular term or object occurs in the multimedia presentation, the index could store the presentation times that should be used to begin playback of the multimedia presentation in the event of a match to the particular term or object.

Fig. 5 is a flowchart illustrating another exemplary process for searching media streams in accordance with another implementation of the invention which does not use indexes. The process of Fig. 5 may be performed in software, firmware, hardware, or any combination thereof. Fig. 5 is described with additional reference to components in Figs. 1, 3 and 4.

A search request and corresponding search criteria are initially received by server 100, step 322. Search engine 270 then compares the data of the media streams corresponding to the search request to the search criteria, step 324. If no match is identified, step 328, then an indication that the search has failed is sent to client 104, step 328. However, if a match is identified, then search engine 270 identifies the temporal location in the media streams corresponding to the match, step 330. Streaming module 205 then begins streaming the media presentation to the client starting at the match location, step 332, analogous to step 316 of Fig. 4.

The comparisons performed by search engine 270 in steps 308 of Fig. 4 and 324 of Fig. 5 can be carried out in a variety of different manners. In one implementation, the comparisons are made through each of the indexes (or media streams) to identify all possible matches to the search criteria and then the temporal location of one of these matches is selected in step 314 (Fig. 4) or 330 (Fig. 5). Alternatively, all of these possible matches may be provided to the user (e.g., the presentation times) and the user can select one of them to "seek" to. Alternatively, as soon as one match of the search criteria is found the comparison can stop and that temporal location seeked to. Alternatively, search engine 270 may identify the current presentation time of the multimedia presentation and search, from that temporal location on, for the next presentation time in each of the media streams that satisfies the search request and select from this set of presentation times.

Recording Data Streams

Returning to Fig. 3, client 104 also includes a stream saving module 272. Module 272 stores composite media stream 202 locally at client 104 as stream 202 is received. Module 272 also optionally receives indexes 260 – 268 and stores them locally at client 104 as well. This storage can be done concurrently with the rendering of the media streams, or alternatively can be carried out independently without rendering the streams. Module 272 can store the media streams locally in response to a user request to store the streams, or alternatively automatically in response to some other event or action (e.g., an indication of search criteria being satisfied from server 100).

Module 272 also generates and stores a markup document that describes how the various media streams are to be rendered (e.g., the screen locations for audio, text, image, and animation streams). This markup document can be generated using any of a variety of conventional programming languages, such as HTML or XML. In the illustrated example, module 272 generates the markup document by modifying a pre-existing markup document, such as one received from server 100 for the rendering of the individual media streams 250 – 258. Module 272 modifies the pre-existing markup document by searching the pre-existing document for references to the locations of media streams 250 – 258 and changing those references to the locally stored media streams. Alternatively, module 272 may generate such a markup document "from scratch".

Module 272 may also "package" the locally stored media streams and the modified markup document into the same file or folder, thereby allowing easier transport of the files. It is to be appreciated that the locally stored media streams could be transferred or copied to another client and played back without requiring further modification to the markup document so long as the path names where the files are stored on the different clients remain the same.

Figs. 6 and 7 are block diagrams illustrating the local storage of a multimedia presentation in accordance with one implementation of the invention. In Figs. 6 and 7, a multimedia presentation is shown including two video streams, an image stream, an audio stream, a text stream, and an animation stream. Initially, these streams are stored at two different remote multimedia servers 340 and 342. A markup document 344 of Fig. 6 at client 104 includes references to or identifiers of the various individual media streams, illustrated as identifiers 346 – 356. As shown, each of the identifiers 346 – 356 identifies one of the remotely stored media streams 358 – 368, respectively.

To store the multimedia presentation locally, the media streams 358 – 368 are streamed (or otherwise copied) to local storage 374 of Fig. 7. Markup document 344 is also modified to generate markup document 376 that references the locally stored media streams 378 – 388. By changing the references to the locally stored media streams 378 – 388, subsequent playback of the multimedia presentation using markup document 376 will result in the locally stored streams 378 – 388 being played back (being input to either demultiplexer 204 or the decoders 222 – 230 directly) rather than the remotely stored streams 358 – 368 of Fig. 6, thereby avoiding any access to remote servers.

Fig. 8 is a flowchart illustrating an exemplary process for recording a multimedia presentation in accordance with one implementation of the invention. The process of Fig. 8 is performed by client 104 of Fig. 3 and may be performed in software, firmware, hardware, or any combination thereof. Fig. 8 is described with additional reference to components in Fig. 3.

Client 104 initially receives a markup document referencing one or more media streams of a media presentation, step 390. Client 104 also receives the media stream(s) of the media presentation, step 392. The received media stream(s) are stored locally at client 104, step 394. Client 104 also modifies the markup document received in step 390 to reference the locally stored media stream(s), step 396, and stores the modified markup document locally, step 398. Client 104 may also optionally package the locally stored media stream(s) and modified markup document into a single file or folder, allowing for easy subsequent transfer of the files.

Conclusion

The invention allows for searching and recording of streaming multimedia data. Any or all of the multiple data streams of a multimedia presentation can advantageously be searched via a single search request by a user, and the user can be immediately presented with the temporal location of the multimedia presentation that satisfies his or her search request. Additionally, a streaming multimedia presentation can advantageously be saved locally, allowing subsequent playback of the presentation when not connected to the remote multimedia servers.

Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as preferred forms of implementing the claimed invention.